REMARKS/ARGUMENTS

Claims 1-54 are currently pending in this application. Claims 1, 10, 13, 18, 26, 27, 33, 39, 40, 42 and 43 have been amended to place the claims in better condition for allowance. In view of the above amendments and following remarks, applicants respectfully submit that the application is in condition for allowance. Entry of this amendment and reconsideration and allowance of the application are therefore respectfully requested.

The Examiner rejected claims 1-7, 9-24, 26-28 and 31-54 under 35 U.S.C. 102(e) as being anticipated by Hervin et al. (U.S. Patent 6,205,560). Applicants respectfully traverse this rejection.

Indepedendent claims 1, 39 and 42 calls for "setting a program counter of the executing service to point to a save stub … setting the program counter of the executing service to point to a restore stub…" Applicants respectfully submit that Hervin et al. does not disclose or sugest the recited limitations.

Rather, the system of Hervin et al. uses a "breakpoint counter that counts down from a predefined number X, such that the exception does not occur until the Xth occurrence of the breakpoint." (Hervin et al., col. 10, lines 18-20). Hervin et al further teaches that by "providing for a breakpoint at the Xth occurrence of a breakpoint equal to the count number …the present invention allows errors that fail to occur in the first occurrence of an instruction to be analyzed and debugged." (Hervin et al., col. 11, lines 1-4). Thus, Hervin et al. merely discloses a counter that counts down from a predefined number X (as counters normally do), such that the exception does not occur until the Xth occurrence of an instruction set.

Hervin et al. does not however, disclose or suggest setting a program counter (i.e. a register containing the next instruction to be executed) to point to a save stub and setting the program counter to point to a restore stub as recited in claims 1, 39 and 42 of the present invention. Applicants therefore respectfully submit that claims 1, 39 and 42 are novel and unobvious over Hervin et al. and are therefore allowable. Applicants further submit that claims 2-12 and 45, claim 51 and claim 53 that depend directly or indirectly from claims 1, 39 and 42 respectively are allowable as are claim 1, 39 and 42 and for additional limitations recited therein.

Independent claims 13, 33, 40 and 43 call for setting "a breakpoint at a last safe point location in an instruction set behind a first breakpoint location if the first breakpoint location is at an unsafe location in the instruction set…" Applicants respectfully submit that Hervin et al. does not disclose or suggest the recited limitations.

Rather, Hervin et al, simply generally refers to setting breakpoints in a set of breakpoint registers, and teaches that the debug registers may be located anywhere within processor…" (Hervin et al., col. 4, lines 32-42). Hervin et al. does not however disclose or suggest setting a breakpoint at a last safe point location in an instruction set if a first breakpoint location is at an unsafe location in the instruction set as recited in claims 13, 33, 40 and 43 of the present invention. Accordingly, applicants respectfully submit that claims 13, 33, 40 and 43 are novel and unobvious over Hervin et al. and are therefore allowable. Applicants further submit that claims 14, 15 and 46, claims 34, 35 and 49, claim 52 and claim 54 that depend directly or indirectly from claims 13, 33, 40 and 43 respectievly are allowable as are claims 13, 33, 40 and 43 and for additional limitations recited therein.

Independent claims 16 and 44 call for "checking for a checksum error within the page of memory; and if the executing service is set to reject the checksum error, saving the page of memory, inserting a breakpoint into the saved page of memory, altering an instruction pointer to the saved page of memory, and processing the saved page of memory." Applicants respectfully submit that Hervin et al. does not disclose or suggest the recited limitations.

Rather Hervin et al. discloses debugging and diagnosing methods which include the use of a logic analyzer connected "to each pin of the processor" and that the "logic analyzer is often used in conjunction with the debug registers 42 when diagnosing and debugging a processor. For example, if the logical analyzer reveals an error in a group of instructions, the instruction addresses are then set as a breakpoints in the debug address registers to further isolate and diagnose the error." (Hervin et al. col. 8, lines 56-67). Hervin et al. thus provides a method for diagnosing errors in an instruction set.

Hervin et al. does not however, disclose checking for a checksum error within a page of memory, and if the executing service is set to reject the checksum error inserting a breakpoint into the saved page of memory, altering an instruction pointer to the saved page of memory, and processing the saved page of memory as recited in claims 16 and 44 of the present invention. Applicants therefore respectfully submit that claims 16 and 44 are novel and unobvious over Hevin et al. and are therefore allowable. Applicants further submit that claims 17 and 47 that depend or indirectly from claim 16 are allowable as is claim 16 and for additional limitations recited therein.

Independent claim 18 recites a system for debugging an executing service on a pipelined CPU architecture comprised in part by a debugger "operable to set a program counter of the executing service to point to a save stub to save a minimum state of the executing service … and to set the program counter of the executing service to point to a restore stub to restore the state of the executing service." Applicants respectfully submit that Hervin et al. does not disclose or suggest the recited limitations.

Rather, as argued above with respect to claim 1, Hervin et al. merely discloses a counter that counts down from a predefined number X (as counters normally do), such that an exception does not occur until the Xth occurrence of an instruction set. Hervin et al. does not however, disclose or suggest setting a program counter (such as a register containing the next instruction to be executed) to point to a save stub and then setting the program counter to point to a restore stub as recited in claim 18 of the present invention.

Applicants therefore respectfully submit that claim 18 recites a novel and unobvious apparatus over Hervin et al. and is therefore allowable. Applicants further submit that claims 19-32 and 48 that depend directly or indirectly from claim 18 are allowable as is claim 18 and for additional limitations recited therein.

Independent claims 36 and 41 recite a system for debugging an executing service on a pipelined CPU architecture which call for means for fetching a page of memory … and if the executing service is set to reject the checksum error means for saving the page of memory, means for inserting a breakpoint into the saved page of memory, means for altering an instruction pointer to the saved page of memory, and means for processing the saved

page of memory." Applicants respectfully submit that Hervin et al. does not disclose or suggest the recited limitations.

Rather, as argued above with respect to claim 16 Hervin et al. provides a method for diagnosing errors in an instruction set. Hervin et al. does not however, disclose means (such as a debugger) for saving a retrieved page of memory with breakpoints inserted if a checksum error is found, altering an instruction pointer to the saved page of memory, and processing the saved page of memory as recited in claims 36 and 41 of the present invention. Applicants therefore respectfully submit that claims 36 and 41 are novel and unobvious over Hevin et al. and are therefore allowable. Applicants further submit that claims 37, 38 and 50 that depend direclty or indirectly from claim 36 are allowable as is claim 36 and for additional limitations recited therein.

It is therefore respectfully submitted that pending claims 1-54 are in condition for allowance, and an early notice of allowance is respectfully requested.

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By _____

Peter A. Nichols
Reg. No. 47,822
626/795-9900

PAN/pan
PAN PAS514932.1-*-07/16/03 4:21 PM